**aBC: An Android Application for Exploring the Proof-of-Work based Blockchain**

By

**Hasibul Hasan Hasib**

Roll: 1907089

&

**Israt Tasnim Esha**

Roll: 1907090



**Supervisor:**

Md. Ahsan Habib

Lecturer

Dept. of Computer Science and Engineering Signature

Khulna University of Engineering & Technology

**Department of Computer Science and Engineering**

**Khulna University of Engineering & Technology**

**Khulna 9203, Bangladesh**

**November 2023**

**Acknowledgements**

First and foremost, I must be grateful to and wish to acknowledge my insightful indebtedness to Md. Ahsan Habib, Lecturer of the Department of Computer Science and Engineering and the supervisor of the project. His unfathomable knowledge in this field influenced me to carry out this project up to this point. His endless endurance, scholarly guidance, continual encouragement, constant and lively supervision, constructive criticism, and priceless suggestions made it possible to come up to this phase. Without his inspiring, enthusiasm, and encouragement, this work could not have been completed. Last, but by no means least, I thank Allah for the talents and abilities I was given that made it possible to undertake this project.

**Authors**

**Abstract**

*WiseChoice*, encapsulated in the *aBC*: An Android Application for Exploring the Proof-of-Work based Blockchain, introduces an Android application designed to address inherent challenges in traditional blockchain systems. Recognizing the lengthy confirmation times and transaction fees associated with mainstream cryptocurrencies like Bitcoin, *aBC* employs a Proof-of-Work-based blockchain for enhanced efficiency. Users enjoy a streamlined process from sign-up to transaction, facilitated by RSA-generated public keys and secure signatures. The mining process is optimized, with added mechanisms ensuring swift and secure block additions. Through these innovations, *aBC* mitigates the obstacles hindering the practical and everyday use of existing blockchain technologies. The project's forward-looking design seeks to make blockchain transactions more accessible, efficient, and user-friendly.

**Contents**

|  |  |  |
| --- | --- | --- |
|  |  | **Page** |
| Acknowledgement |  | ii |
| Abstract |  | iii |
| Contents |  | iv |
| List of Tables |  | vi |
| List of Figures |  | vii |

|  |  |  |
| --- | --- | --- |
| **1** | **Introduction** | 1 |
|  | 1.1 Background | 1 |
|  | 1.1.1 Key Challenges in Traditional Transaction  1.1.2 Challenges in Existing Blockchain Solutions  1.1.3 How abc Overcomes Existing Blockchain Challenges | 1  2  2 |
|  | 1.2 Objectives | 3 |
|  | 1.3 Scopes | 4 |
|  | 1.4 Unfamiliarity of the Problem | 4 |
|  | 1.5 Project planning | 5 |
| **2** | **Related Works** | 6 |
|  | 2.1 Existing solutions | 6 |
|  | 2.2 Contributions And Limitations In Existing Solutions | 8 |
| **3** | **System Design** | 9 |
|  | 3.1 Analysis of The System | 9 |
|  | 3.2 System Architecture | 11 |
|  | 3.3 Tools Used | 15 |
|  | 3.3.1 Android Studio | 15 |
|  | 3.3.2 Kotlin | 15 |
|  | 3.3.3 Firebase | 16 |
|  | 3.3.4 RSA Algorithm | 16 |
| **4** | **Project Implementation** | 17 |
|  | 4.1 System Implementation | 17 |
|  | 4.1.1 Experimental Analysis  4.1.2 User Manual | 17  19 |
|  | 4.2 Morality Or Ethical Issues | 22 |
|  | 4.3 Socio-Economic Impact And Sustainability | 22 |
| **5** | **Conclusions** | 22 |
|  | 5.1 Conclusion And Challenges Faced | 22 |
|  | 5.2 Future Work | 23 |
|  | **References** | 23 |
|  |  |  |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Description** | **Page** |
| 2.1 | Contributions And Limitations in Existing Solutions | 9 |
| 4.1 | System Analysis | 17 |
| 4.2 | System Features Comparison | 18 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page** |
| 1.1 | Project Planning Of *aBC* | 6 |
| 3.1 | Data Flow Diagram of The Project | 10 |
| 3.2 | Activity Diagram of The Project | 10 |
| 3.3 | UML Diagram of The Project | 11 |
| 3.4 | Architecture of The System | 12 |
| 4.1 | System Performance Analysis | 17 |
| 4.2 | User Register to Account and Login with Registered Information. | 19 |
| 4.3 | User Transacts Money, Miner Verifies Transactions and Mines Blocks | 20 |
| 4.4 | Miner Verifies Transactions and Accept Block, User Updates Account | 21 |

1. Introduction
   1. Background

Conventional transaction systems have long served as the bedrock of financial operations, relying on centralized structures and intermediaries. In these conventional systems, clients deposit funds, and transactions are facilitated by banks, often leading to delays and operational costs due to the involvement of multiple parties. The traditional transaction paradigm revolves around maintaining central ledgers, updated with every financial transaction, and necessitates each member of the business network to possess their own ledger copy. Unfortunately, this system is not without notable drawbacks, including inefficiencies in transaction processing, high maintenance costs, and susceptibility to fraud or cyber-attacks [1].

Alongside traditional transaction, existing blockchain systems have emerged as alternatives, offering decentralization and heightened security. However, blockchain systems are not immune to challenges such as scalability issues, energy consumption concerns, and the complexity of consensus mechanisms pose significant obstacles [2]. While decentralized, existing blockchain solutions face hurdles in achieving the streamlined efficiency required for everyday transactions.

* + 1. **Key Challenges i**n Traditional Transaction

To understand the complexities, let's discuss the primary issues: centralization, trusted party, efficiency, maintenance costs, and vulnerability.

* Centralization: In centralized systems, if the main server or authority has an issue, the entire system can fail. Plus, everyone has to follow the rules set by that central authority, which might not be fair for everyone.
* Trusted Party: In centralized systems, we trust one main party to make sure everything works. But if this trusted party makes a mistake or does something wrong, it can cause big problems. Also, because everything is controlled by this party, we might not always know if they're doing things the right way.
* **Efficiency:** Traditional banking processes involve multiple intermediaries, leading to increased overheads and the need for substantial human resources to handle transactions.
* **Maintenance Costs:** Every transaction in traditional banking incurs maintenance charges, and internal audits contribute to prolonged processes.
* **Vulnerability:** Traditional banking systems are susceptible to fraud and malicious ledger modifications, posing a significant risk in the face of cyber-attacks.
  + 1. Challenges in Existing Blockchain Solutions

While existing blockchain solutions offer decentralization and security, they are not without challenges.

* **Decentralization and Security:** Existing blockchain solutions have earned praise for their decentralized structures and strong security features. This means that no single person or group has all the control, making it robust and secure. However, these strengths also bring challenges that need careful attention. [2].
* **Scalability Concerns:** One big challenge for current blockchains is scalability – how well they can handle more and more transactions It's a bit like playing a game where, as more friends join, it slows down. Current blockchains face similar issues, struggling to keep up as more transactions happen. It's like needing a bigger playground as more friends join the game[2].
* **High Energy Consumption, Especially in Proof-of-Work:** Another challenge is that some blockchains use a lot of energy, especially those using Proof-of-Work to decide who gets to add new information [2]. It's like a super hungry monster needing a lot of energy. This isn't great for the environment, and we're trying to find smarter ways to do it – like teaching our monster to eat more efficiently.
  + 1. How *aBC* Overcomes Existing Blockchain Challenges

Here is how *aBC* tackles each of these challenges.

* **Five Layers of Security:** At first when a new account added 5 random account is checked for verifying and adding the previous transactions to the account. Secondly when a transaction occurs data of the accounts also checked. After that when verifying transactions data of the transactions are verified decoding signature. Fourthly when mining a block, transactions in the block will be verified and finally when accepting the block transactions will be verified using public key and the signature.
* **Efficiency Enhancement**: *aBC* optimizes the Proof-of-Work mechanism, ensuring that transactions are processed with increased efficiency, minimizing delays and reducing resource consumption.
* **Cost Reduction**: By streamlining the mining process and enhancing transaction validation, *aBC* aims to lower maintenance costs associated with blockchain transactions.
* **Security Fortification**: The project introduces advanced security measures to mitigate vulnerabilities, ensuring that transactions remain resilient against fraud and cyber-attacks.

Through these innovations, *aBC* emerges as a dynamic solution that not only addresses the inefficiencies of traditional banking but also strategically tackles the challenges inherent in existing blockchain systems. The project envisions a future where blockchain technology is seamlessly integrated, providing a secure, efficient, and user-friendly platform for everyday transactions.

* 1. Objectives

The main objectives of the project *aBC* are –

* **Overcome Conventional Blockchain Limitations**: Meticulously design *aBC* to surpass constraints in traditional blockchain systems.
* **Central Adoption of Proof-of-Work**: Embrace Proof-of-Work as a strategic cornerstone to enhance transaction efficiency.
* **Diminish Confirmation Times**: Reduce transaction confirmation times for a more efficient and responsive experience.
* **Alleviate Associated Fees**: Implement strategies to alleviate fees linked with transactions, ensuring cost-effectiveness.
* **Detailed Roadmap Development**: Create a comprehensive roadmap guiding the development and implementation of *aBC*.
* **Disciplined Focus on Objectives**: Maintain a disciplined focus on outlined objectives throughout the project's lifecycle.
* **Set New Standards in Blockchain**: Aspire to establish new benchmarks for streamlined, cost-effective, and swift blockchain transactions.
* **Redefined User Experience:** Redefine user experience by introducing innovations that streamline transactions.
* **Prioritize Cost-Effectiveness**: Prioritize cost-effectiveness through strategic measures to make *aBC* economically viable.
  1. Scope

*aBC* aims to do a lot of things for users, especially making blockchain easy to use. It focuses on being simple, friendly, and efficient, going beyond the usual to give people a better experience with technology

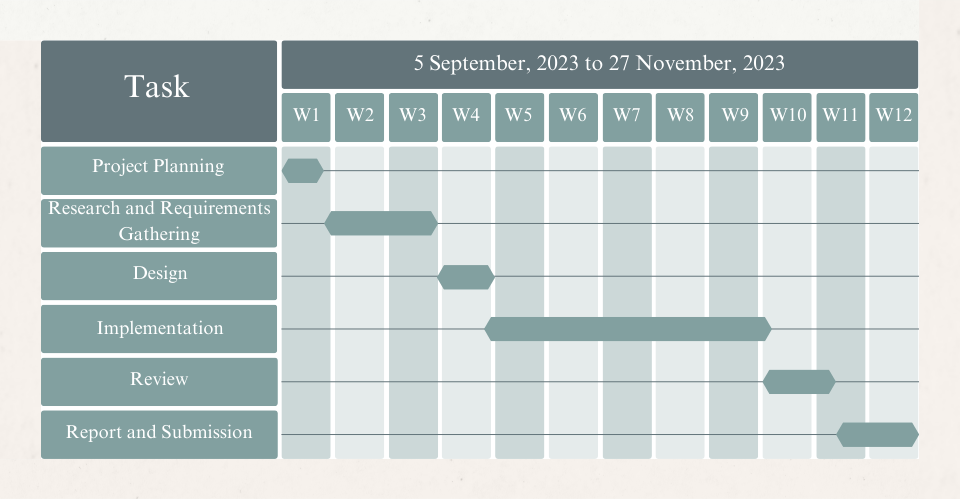
* **The Wide App Coverage:** *aBC* covers many aspects and features for users.
* **Focus on Proof-of-Work:** It mainly works with a Proof-of-Work blockchain, making things efficient.
* **User-Friendly Features:** It has easy-to-use features to make it simple for users.
* **Clearly Defined Limits:** There are clear rules and limits for how *aBC* works.
* **Intended Impact on Users:** It aims to have a positive effect on users and the blockchain community.
* **Innovative Approach:** It goes beyond the usual ways of doing things, trying new and smart solutions.
* **Better User Experiences:** It promises to make using the app a better experience for users.
* **Smoother Transactions:** It focuses on making transactions smooth and easy.
* **Making Blockchain Easier:** It wants to set a new standard for making blockchain technology easier to use.
* **Efficiency is Key:** The main goal is to make *aBC* work really well and be at the front of blockchain technology.
  1. Unfamiliarity of The Problem

In the realm of money apps, *aBC* stands out by venturing into uncharted territory, redefining the landscape by prioritizing not just security, but also speed and ease in your financial transactions.

* **Unique Challenge in Money Apps**: *aBC* steps into a different challenge compared to other money apps.
* **Different Focus from Regular Apps:** While most apps deal with regular banking issues, *aBC* takes a unique route.
* **Exploring New Paths:** It explores a less-travelled path, ensuring super secure and quick transactions.
* **Beyond Basic Money App Functions**: Unlike other apps handling basics, this project combines blockchain tech with user-friendly features.
* **Creating a New Road:** It's like building a new road in the world of financial tech for safer and hassle-free money moves.
* **Fresh Perspective in Financial Tech:** *aBC* brings a fresh perspective, solving problems others might have missed.
* **Aim for Smoother Financial Journey:** It aims to make your financial journey smoother than ever by addressing unique challenges.
  1. Project Planning

The meticulous planning of *aBC* project involved a systematic approach to ensure its success. Figure 1.1 illustrates the key milestones and stages in the project's timeline.

* Project Planning: Initiated the project with a thorough conceptualization phase, defining core objectives, user requirements, and *aBC*’s unique value proposition. Shaped conceptual ideas in the design and architecture phase, making crucial decisions on user interface, system architecture, and database structures for a robust foundation.
* Researches and Requirements Gathering: Carefully selected tools like Android Studio, Kotlin, Firebase, and RSA algorithm, ensuring compatibility and efficiency aligned with project goals.
* Design and Implementation: Executed coding and implementation of *aBC*, integrating the Proof-of-Work mechanism, RSA algorithm, and crafting a user-friendly interface.
* Review: Rigorously tested *aBC*, simulating various scenarios to validate the effectiveness of Proof-of-Work and RSA algorithm in real-world situations.



**Figure 1.1:** Project Planning of *aBC*

* Report and Submission: Released *aBC* to users in the deployment phase, establishing feedback mechanisms for continuous improvement. Continuous improvements and updates iteratively based on user feedback and emerging trends in blockchain technology.

This structured and progressive project planning process, illustrated in Figure 1.1, exemplifies the meticulous journey of *aBC* from conceptualization to a fully functional, user-centric Proof-of-Work-based blockchain application.

1. Related Works
   1. Existing Solutions

* **Conditional Anonymous Payment System DCAP[3]**: In this proposal, a blockchain-based conditional anonymous payment system is introduced. Operated on a private Ethereum chain, a central certificate authority (CA) issues certificates to users. The system utilizes smart contracts for transactions, maintaining privacy through a dynamic-address approach. However, the reliance on a central authority introduces a bottleneck and a single point of failure, impacting system resiliency. Additionally, the account models lack the security of UTXO models and struggle with parallel transaction processing. Here average Tx size is 548 Byte, Tx generation time 57.02 ms, Tx verification time 32.43 ms.
* **Auditable Decentralized Confidential Payment System PGC[4]**: The authors present an auditable decentralized confidential payment system using lightweight zero-knowledge (ZK) proofs. While these proofs address scalability and low throughput issues, the lack of details about the underlying blockchain and consensus algorithm raises transparency concerns. ZK proofs, although lightweight, encrypt transacting amounts, hindering auditability. Users can only prove their total funds through ZK proofs, introducing reliability challenges susceptible to dishonest users. Here average Tx size is 1408 Byte, Tx generation time 28 ms, Tx verification time 9 ms.
* **Auditable Privacy-Preserving UTXO-based Cryptocurrency FAPC[5]**: This proposal suggests a privacy-preserving UTXO-based cryptocurrency incorporating ring signatures and long-term traceable addresses. However, critical details about the blockchain implementation are omitted. Despite providing guaranteed privacy, the reliance on ring signatures contributes to scalability and low throughput issues due to heavyweight transactions and increased computational complexity. Here average Tx size is 548 Byte, Tx generation time 57.02 ms, Tx verification time 32.43 ms.
* **Privacy-Preserving Token Management System [6]:** The authors propose a privacy-preserving token management system built on Hyperledger Fabric. To enable auditability, transaction information is encrypted using the public keys of auditors. However, this introduces a single point of failure as receivers rely on auditors to decrypt transactions. The process of transferring tokens involves certifiers, adding complexity and compromising system resiliency. Overall, the system exhibits high computational complexity and lacks transparency. Here average Tx size is 64512 Byte, Tx generation time 1992.84 ms, Tx verification time 2885.13 ms.
* **Intelligent Cross-Border Transaction System [7]:** This work introduces an intelligent cross-border transaction system based on a consortium blockchain with smart contracts. Despite numerical experiments, details about the transaction verification scheme are unclear, raising concerns about privacy and transparency. The proposed system lacks clarity on how it addresses cross-border payment challenges on a global scale.
* **Compact-Sized Anonymous and Auditable Distributed Payment System [8]:** The authors propose a compact-sized anonymous and auditable distributed payment system where banks transact via Mini Ledger. While reducing memory consumption, the method faces challenges in guaranteeing transparency. The hash value's insufficiency for non-interactively verifying a transaction, coupled with the deletion of original transaction data, poses potential transparency issues in maintaining data integrity and privacy.

In summary, these works exhibit compromises in providing a comprehensive solution encompassing auditability, privacy, transparency, and resiliency. The discussions also highlight the oversight in considering cross-border payment challenges from a global perspective. transactions.

* 1. Contributions and Limitations of Existing Solutions

By analyzing these existing solutions, it's evident that none provide a comprehensive solution to the challenges of auditability, privacy, transparency, and resiliency simultaneously. Additionally, none consider the specific context of cross-border payment from a global perspective.

**Table 2.1**: Contributions and Limitations in Existing Solutions

|  |  |  |
| --- | --- | --- |
| Works | Contributions | Limitations |
| [3] | * Privacy through dynamic-address approach * -Utilizes Ethereum smart contracts | * Central authority introduces a single point of failure * - Account models less secure than UTXO models |
| [4] | * - Lightweight ZK proofs for scalability | * Lack of details on blockchain and consensus algorithm * ZK proofs hinder amount observation |
| [5] | * Guaranteed privacy through ring signatures | * Omitted details on blockchain implementation * Heavyweight transactions impact scalability |
| [6] | * Built on Hyperledger Fabric for privacy | * Single point of failure in decryption process * Complexity in token transfer process |
| [7] | * Introduction of intelligent cross-border transactions | * Unclear transaction verification scheme * Lack of details on privacy and transparency |
| [8] | * Reduction in memory consumption | * Insufficiency of hash value for non-interactively verifying transactions * Potential transparency issues in data integrity and privacy |

This analysis emphasizes the need for a novel solution, as the identified problem is not directly addressed in current literature.

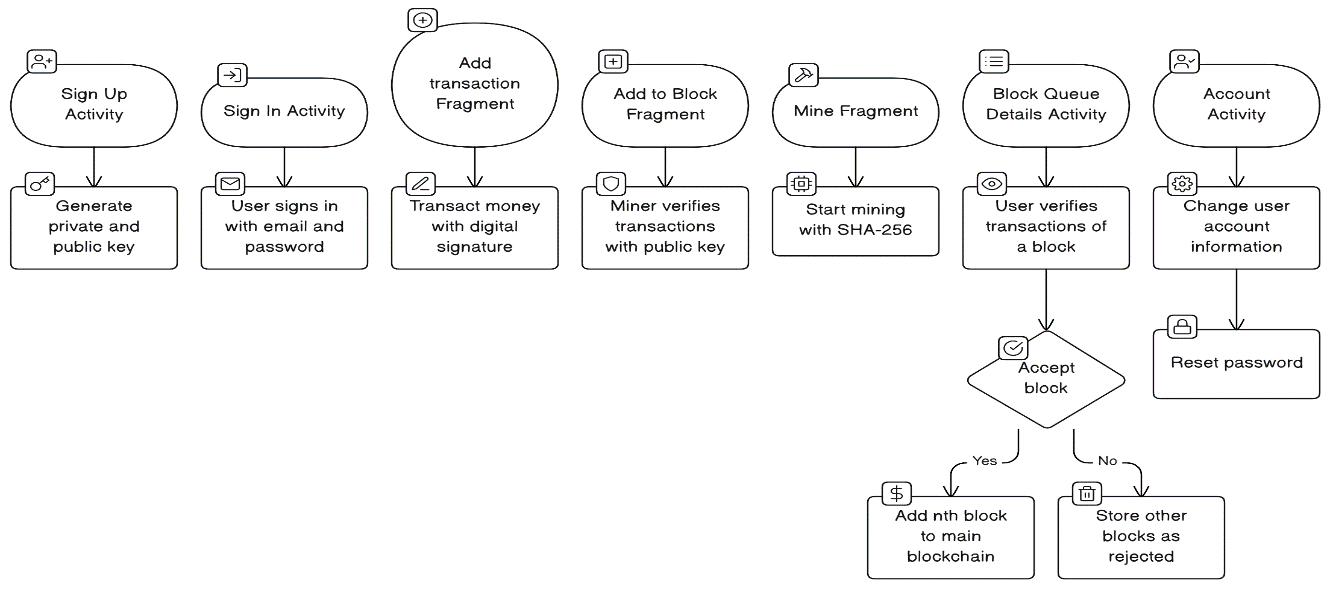
1. System Design
   1. Analysis of The System

Figure 3.1 and in Figure 3.2 we can see that Users sign up with private and public keys.

Transactions include fees and digital signatures. Miners verify and store transactions, mine

using SHA-256, and users validate blocks. Accepted blocks result in miners and recipients

receiving funds, ensuring secure and transparent transactions on the blockchain.



Sign Up

Sign In

Add Transaction

Mine

Add

To Block

Block

Details

Account

**Figure 3.1:** Data flow diagram of the project

Change Information

Verify Transactions

Start Mining

Verify Transaction

Transact Money

Authenti-cation

Generate key

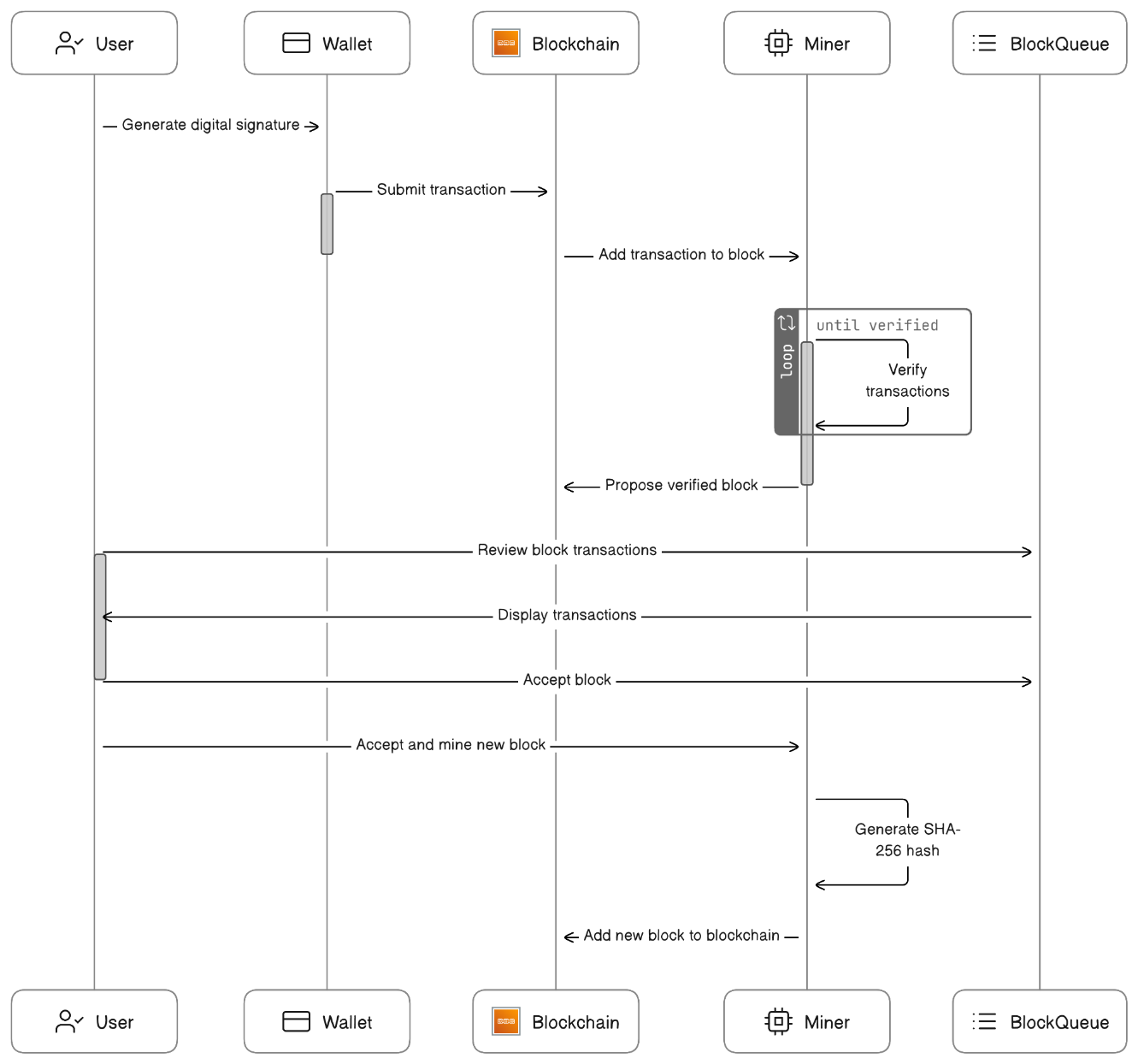
Accept Block

Reset Password

Rejected block

Added to Blockchain

**Figure 3.1:** Data Flow Diagram of The Project



Generate Digital Signature

Block

Chain

Block Queue

Miner 

Wallet

User

Submit Transaction

Add Transaction to Block

Verify Transactions

Propose Verified Block

Display Transactions

Review Block Transactions

Accept Block

Accept and Mine New Block

Generate SHA-256 Hash

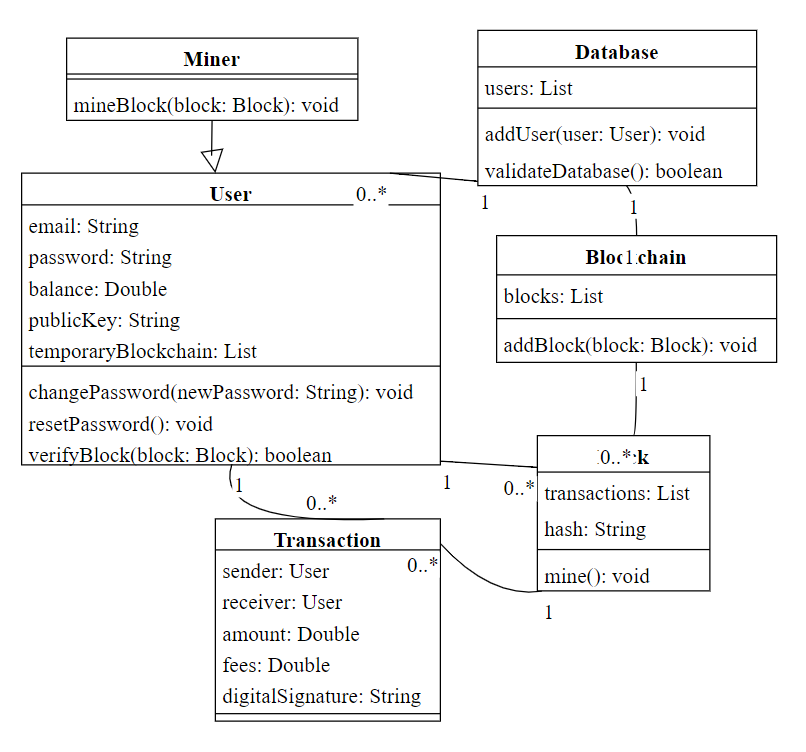
Add New Block To Blockchain

**Figure 3.2:** Activity Diagram of The Project.

* 1. System Architecture

Here's how the system architecture works:

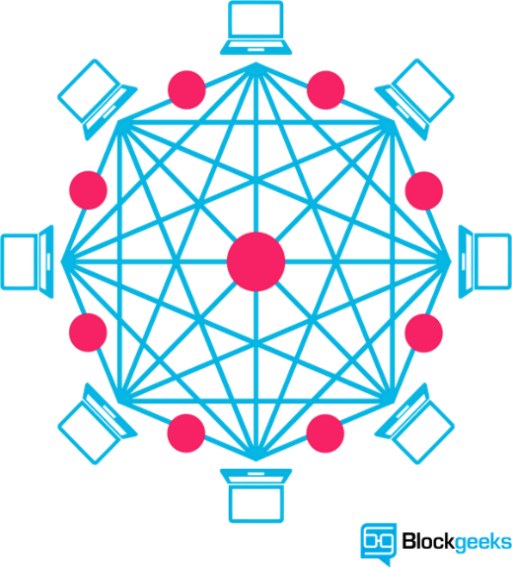
* **User Accounts**: Users sign up and log in using their email and password. Each user starts with $100 in their digital wallet.
* **Sending Money:** When a user wants to send money to someone else, the system checks if the receiver exists and if the sender has enough money.



**Figure 3.3:** UML Diagram of The Project

If all good, the sender's money is reduced, a small fee is added for the system's "miners," and the transaction is sent to everyone in the system.

* **Digital Signatures:** Every transaction gets a special digital signature, like a unique stamp, to make sure it's authentic and can't be tampered with.
* **Mining and Verification:** Miners are like digital detectives. They verify a bunch of transactions using a special math process called SHA-256. Once they've verified enough transactions, they start mining. Mining involves solving complex puzzles using SHA-256.The mined block (a group of transactions) is sent to everyone's "block queue."
* **User Verification and Acceptance:** Users check the transactions in a block and decide if they're legit. If yes, they accept the block, and the transactions go into their temporary blockchain.



Send Money

Sender

Receiver

Amount

Miner’s Fees

Attacker

Security

Attack

Mining

Fees



Reward

Miner

Receive Money



**Figure 3.4:** Architecture of The System.

* **Adding to the Main Blockchain:** If a user accepts a block, they can start mining the next one. Once mined, the new block is added to the main blockchain, updating everyone's block queue and temporary blockchain.
* **Money Distribution:** When a block is added, the receiver and miner get the money from the transactions. The miner also gets an extra $5 as a reward for their hard work.
* **Transparency:** All money transfers are shown to everyone. This transparency helps catch anyone trying to mess with the system.
* **Database Security:** When a new user signs up, their information is compared with four random users to make sure everything matches. This adds an extra layer of security

Algorithms:

|  |
| --- |
| **Algorithm 1** Registration of a user |
| **Input:** Name *N*, Phone Number *PH*, Email *E*, Password *PW*  **Output**: *PUB\_KEY, PRIV\_KEY*  *AUTH* ← initAuth()  createUser(*AUTH*, *E*, *PW*)  **if** success(*AUTH*) **then**  *USER* ← getUser(*AUTH*)  *KEYS* ← genKeyPair()  *PUB\_KEY* ← getPubKey(*KEYS*)  *PRIV\_KEY* ← getPrivKey(*KEYS*)  *PRIV\_KEY* ← toBase64(*PRIV\_KEY*.encoded)  **if** noMiners() **then**  saveUser(*N*, *PH*, *PUB\_KEY*, *PRIV\_KEY*)  **else if** atLeastOneMiner() **then**  *MINERS* ← getRandomMiners()  **if** validateIntegrity(*MINERS*) **then**  saveUser(*N, PH, PUB\_KEY, PRIV\_KEY*)  goToSignIn() ∧ finishSignUp()  **else**  err("Corrupted Data") ∧ toast("Corrupted Data")  **end if**  **end if**  **else**  error("Auth Failed") ∧ toast("Auth Failed")  **end if** |

|  |
| --- |
| **Algorithm 2** Add Transaction |
| **Input:** Receiver *R*, Amount *A*, Fees *F*, Sender *S*, *USER*  **Output**: Transaction *T*, Signature *SIG*  fetchKey(*USER*)  **if** keyExists() **then**  *T* ← *R+A+F*+getCurrentDateTime()  *SIG* ← signData(*PRIVATE\_KEY, T*)  toast("Signature created")  **if** minerHasKey() **then**  performTransaction(*SIG*)  reduceMoney(*USER, A+F*)  **else**  toast("No private key found")  **end if**  **else**  toast("Invalid data input")  **end if** |

|  |
| --- |
| **Algorithm 3** Add to Block |
| **Input:** *TRANS*  **Output:** *TRANS*.*STS*  DATA ← decodeData(*PUBLIC\_KEY*, *TRANS*.SIG)  **if** *TRANS*.*STS*==”Unrecognized” **then**  **if** match(*DATA*, *TRANS.S*, *TRANS.R*, *TRANS.A*, *TRANS.F*) **then**  *TRANS*.*STS* ← “Verified”  addToBlock()  **else**  *TRANS*.*STS* ← “Not Verified”  **end if**  **end if** |

|  |
| --- |
| **Algorithm 4** Mining Process |
| **Input:** *TRANS*[], *PR\_BLOCK*  **Output:** *BLOCKCHAIN\_BLOCK, BLOCK\_QUEUE\_BLOCK*  **for** i in range (0, len(*PR\_BLOCK*.*TRANS*[])) **do**  *PR\_DATA += PR\_BLOCK.TRANS[I].S, PR\_BLOCK.TRANS[I].R, PR\_BLOCK.TRANS[I].A, PR\_BLOCK.TRANS[I].F*  **end for**  **if** match(*PR\_BLOCK*.*HASH*, RSA*\_*256(*PR\_DATA*)) **then**  **for** i in range (0, count(*TRANS*)) **do**  *DATA += TRANS[I].S, TRANS[I].R, TRANS[I].A, TRANS[I].F*  **end for**  *BLOCK*.*HASH* ← RSA*\_*256(*DATA*)  startMining(*BLOCK*) ∧ blockChained*(PR\_BLOCK*)  **for** i in range (0, len(*PR\_BLOCK*.*TRANS*[])) **do**  *PR\_BLOCK.TRANS[I].STS* ← “Blocked”  *PR\_BLOCK.TRANS[I].BLOCKNO* ← *BLOCK.BLOCKNO*  **end for**  blockQueued(*BLOCK*)  **for** i in range (0, len(*PR\_BLOCK.TRANS*[])) **do**  sendToReceiver(*PRBLOCK.TRANS[I].R, PRBLOCK.TRANS[I].A*)  sendToMiner(*PRBLOCK.MINER, PRBLOCK.TRANS[I].F*)  **end for**  **else**  err("Corrupted Data") ∧ toast("Corrupted Data")  **end if** |

|  |
| --- |
| **Algorithm 5** Block Queue and Acceptance |
| **Input:** *BLOCK*  **Output:** *VERIFIED\_BLOCK*  **for** i in range (0, len(*block*.*trans*[])) **do**  *DATA ←* decodeData*(PUBLIC\_KEY, TRANS[I].SIG)*  **if** match(*DATA, TRANS[I].S, TRANS[I].R, TRANS[I].A, TRANS[I].F*) **then**  *TRANS[I].STS* ← “Verified”  *CNT++*  **else**  toast("Corrupted Data")  **end if**  **if** *CNT==* len*(BLOCK.TRANS[])* **then**  acceptBlock()  **end if** |

3.3 Tools used

In the development of the *aBC* project, a suite of robust tools was meticulously selected to ensure efficiency, functionality, and security. The primary tools employed in this endeavor include:

**3.3.1 Android Studio**

Android Studio served as the principal integrated development environment (IDE), offering a comprehensive toolkit for designing, coding, and testing the *WiseChoice* Android application. Its versatility and compatibility with Android platforms facilitated the creation of a seamless and user-friendly interface.

**3.3.2 Kotlin**

Kotlin, a modern programming language endorsed by Google for Android app development, played a pivotal role in crafting the *WiseChoice* application. Renowned for its conciseness and expressiveness, Kotlin expedited the coding process, enhancing the overall development experience.

**3.3.3 Firebase**

Firebase, a comprehensive mobile and web application development platform, was instrumental in implementing various functionalities within *WiseChoice*. We implemented user authentication using Firebase Authentication and utilized the Firebase Realtime Database for storing and retrieving transaction and block data. To enhance security, we established distinct instances for each account, storing all transaction and blockchain details independently on each instance. This approach ensures a more secure system architecture by isolating data for individual user accounts.

**3.3.4 RSA Algorithm**

The RSA algorithm, a cornerstone of modern cryptography, was integrated into *WiseChoice* to ensure the secure generation of public and private keys. Employing RSA added an additional layer of protection to user accounts and transactions, aligning with the project's commitment to robust security measures.

**Key Generation:**

1. Choose two large prime numbers, *p* and *q*.
2. Compute ***n*= *p \* q***.
3. Compute ***ϕ*(*n*)=(*p*−1)(*q*−1),** where *ϕ* is Euler's totient function.
4. Choose a public exponent *e* such that **1<*e*<*ϕ*(*n*)** and *e* is coprime to ***ϕ*(*n*).**
5. Compute the private exponent *d* such that ***d*≡*e*−1mod*ϕ*(*n*).**
6. The public key is (*n*,*e*), and the private key is (*n*,*d*).

**Signature Creation:**

1. The sender uses their private key (*n*,*d*) to create a digital signature for a message ***S*≡*Md* mod *n***.

**Signature Verification:**

1. The recipient uses the sender's public key (*n*,*e*) to verify the signature

***M*′≡*Se* mod *n*.**

1. If ***M*′=*M*,** the signature is valid.
2. Project Implementation

**4.1 System Implementation**

4.1.1 Experimental Analysis:

Table 4.1 shows the average time of signup, sign in, transaction time, validity check time, mining time of the different nodes while performing tasks. Table 4.2 gives a clear comparison of different systems highlighting Tx generation time, mining time, address type, privacy, anonymity etc. Figure 4.1 illustrates time comparisons for different tasks in the project.

Table 4.1: System Analysis

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Experiment Name | Attempt 1 | Attempt 2 | Attempt 3 | Attempt4 | Attempt 5 | Average Time Required |
| Sign Up | 5.17 | 4.55 | 3.85 | 3.20 | 3.15 | 3.984 |
| Sign In | 3.70 | 2.30 | 2.40 | 2.40 | 2.30 | 2.183 |
| Transaction Time | 2.30 | 2.20 | 2.30 | 1.77 | 1.88 | 2.090 |
| Validity Check Time | 1.20 | 1.00 | 0.85 | 0.60 | 0.30 | 0.791 |
| Mining Time | 2.45 | 9.30 | 11.60 | 1.20 | 0.83 | 5.076 |

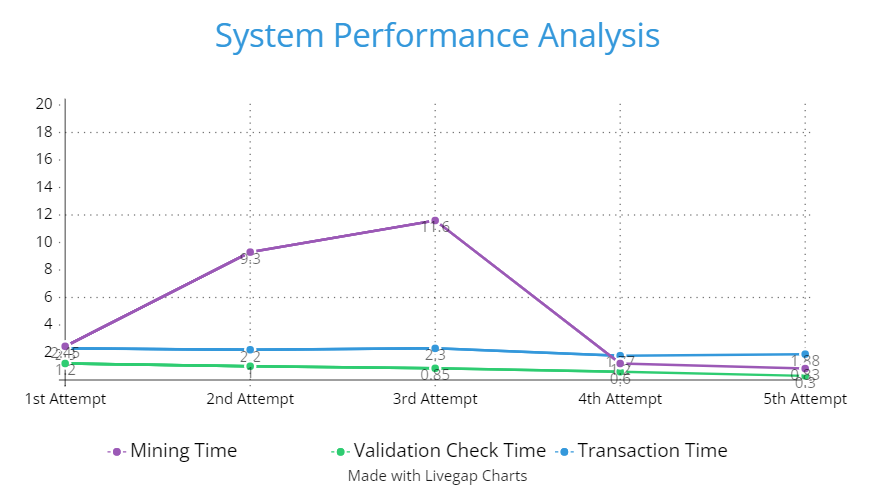


Figure 4.1: System Performance Analysis

Table 4.2: Systems Features Comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameters | PGC[4] | FAPC[5] | [6] | DCAP[3] | Proposed System |
| Transaction Type | Account-based | UTXO-based | UTXO-based | Account-based | UTXO-based |
| Network Type | - | Decentralized | Partially | Partially | Decentralized |
| Blockchain Used | - | - | Hyperledger  Fabric | Ethereum | Own |
| Consensus | - | - | - | PBFT | PoW |
| Anonymity | No | Yes | Yes | Yes | Yes |
| Privacy | Yes | Yes | Yes | Yes | Yes |
| Transparency | Semi Transparent | Fully Transparent | Not Transparent | Semi Transparent | Fully Transparent |
| Auditability | Yes | Yes | Yes | Yes | Yes |
| Authentication Type | - | - | Interactive | Interactive | Password-Based. |
| Resiliency | - | - | No | No | Yes |
| Loss Recovery Scope | - | No | Yes | No | Yes |
| Address Type | Actual Address | Id-linked | Anonymous Credentials | Id-linked | Actual Address |
| Verification Scheme | ZK proofs | Ring Signatures | ZK proofs | ZK proofs | Digital Signature |

4.1.2 User Manual

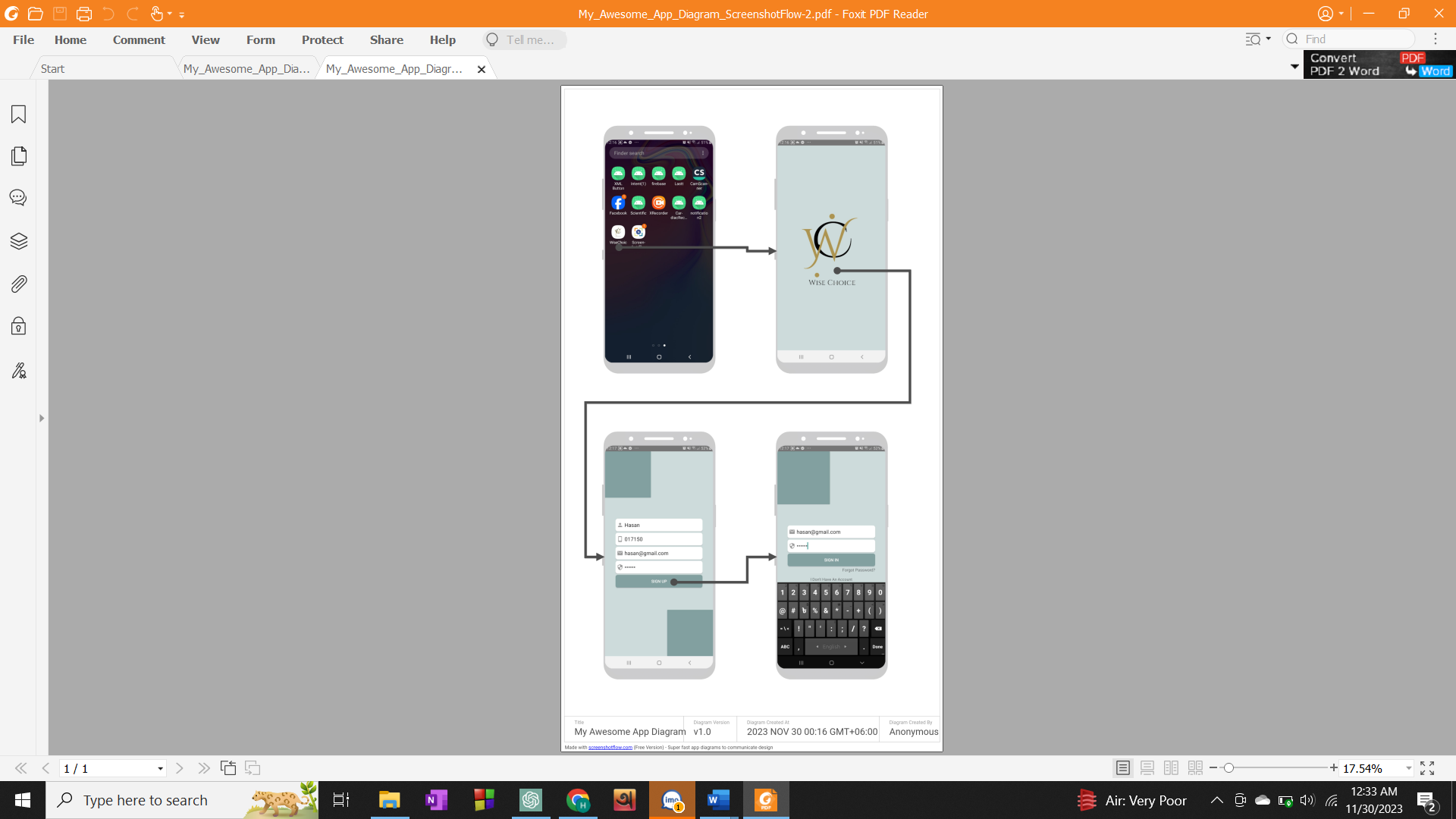


Figure 4.2: User Register to Account and Login with Registered Information.

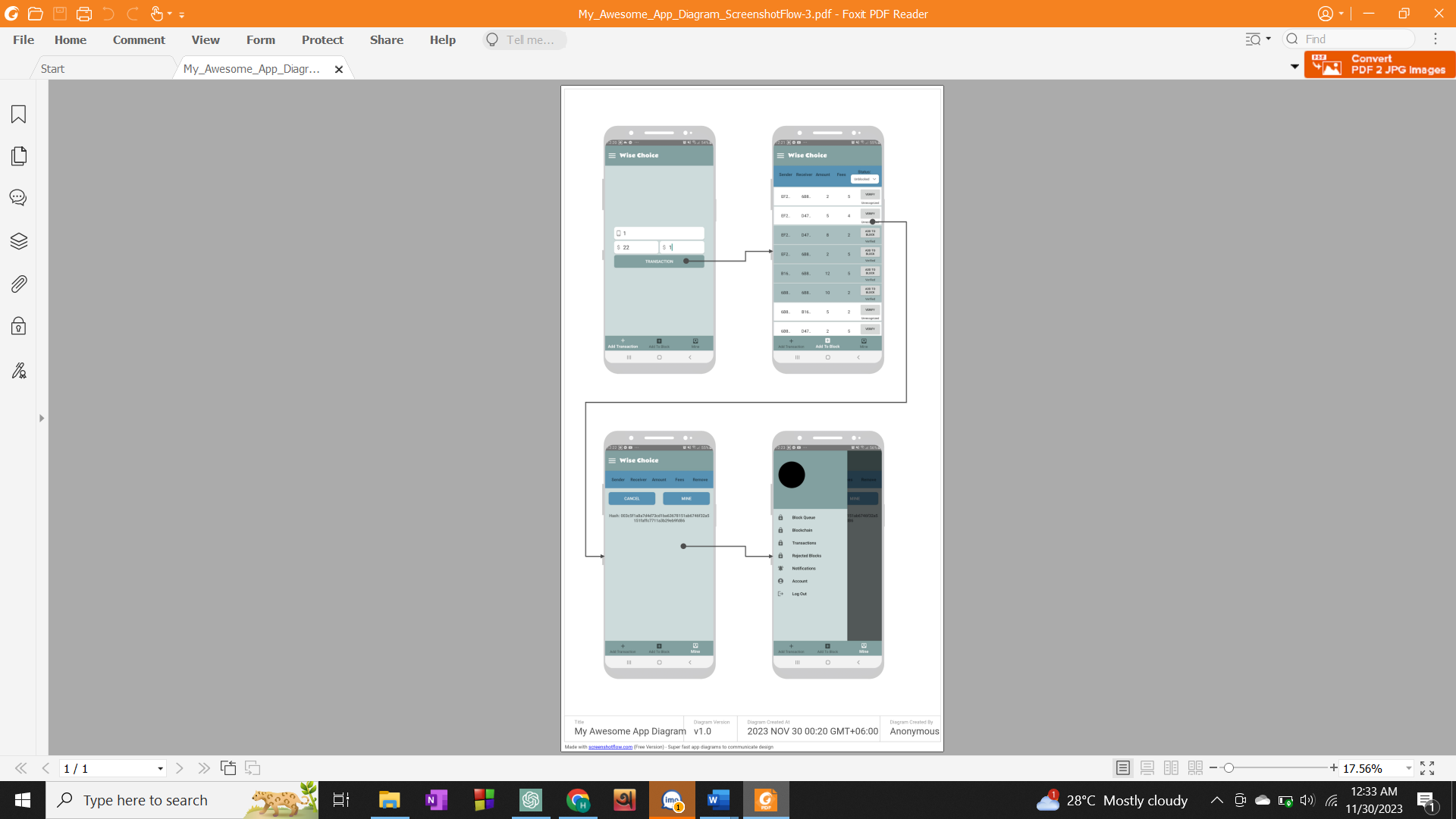


Figure 4.3: User Transacts Money, Miner Verifies Transactions and Mines Blocks



Figure 4.4: Miner Verifies Transactions and Accept Block, User Updates Account

The provided figures 4.1, 4.2 and 4.3 outlines the sequential steps that users will follow to navigate through the app.

4.2 Morality or Ethical Issues

Implementing the blockchain-based transaction system requires addressing ethical considerations, including user privacy, security risks, transparency vs. anonymity, fair mining practices, and ensuring informed consent. Proper citations and acknowledgments are crucial to avoid plagiarism.

4.3 Socio-Economic Impact and Sustainability

The project positively impacts financial inclusion, reduces transaction costs, enhances security and trust, and considers cultural and legal aspects. Environmental implications of the mining process should be addressed for long-term sustainability. The system contributes to societal health and safety by preventing fraud.

5 Conclusion

5.1 Conclusion and Challenges Faced

In conclusion, the implemented transaction system with its own blockchain demonstrates a secure and transparent method for handling financial transactions. The use of digital signatures, SHA-256 hash generation, and decentralized verification processes ensures the integrity and authenticity of each transaction. The incorporation of mining as a consensus algorithm enhances the security of the system and incentivizes participants.

Challenges faced during the development and implementation of the system included:

* **Security Concerns**: Ensuring the security of user data, private keys, and transactions is a constant challenge. Ongoing efforts must be made to stay ahead of potential vulnerabilities and attacks.
* **Scalability:** As the number of users and transactions increases, the system's scalability may become a concern. Future optimizations and enhancements should be explored to handle a growing user base efficiently.
* **User Experience**: While the system prioritizes security, the user experience is crucial for widespread adoption. Striking a balance between security measures and user-friendly interfaces remains an ongoing challenge.

Despite these challenges, the implemented system provides a robust foundation for secure and transparent financial transactions.

5.2 Future Study

In future studies and developments could focus on the following areas:

* **Smart Contracts:** Integration of smart contracts could automate and enforce predefined rules for transactions, adding programmability to the blockchain.
* **Privacy Enhancements:** Exploring privacy-focused technologies, such as zero-knowledge proofs, can enhance user privacy while maintaining the transparency of transactions.
* **Interoperability:** Investigating ways to enable interoperability with other blockchain systems and traditional financial networks can expand the reach and utility of the implemented system.

# Governance Models: Implementing decentralized governance models can empower users to participate in decision-making processes related to the evolution of the blockchain system.

# Research on Quantum-Resistant Algorithms: As quantum computing advances, research into quantum-resistant algorithms can ensure the continued security of the blockchain against emerging threats.

**References**

[1]L. Kessem, “Mobile banking trojans as keen on cryptocurrency as pc malware,” Mar

2018. [Online]. Available: <https://securityintelligence.com/mobile-banking-trojans->

as-keenon-cryptocurrency-as-pc-malware/

[2] “Owasp mobile security project.” [Online]. Available:

https://www.owasp.org/index.php/Mobile Top 10 2016-Top 10

[3] C. Lin, D. He, X. Huang, M. K. Khan, and K. R. Choo, “DCAP:A secure and

efficient decentralized conditional anonymous payment system based on

blockchain,” IEEE Trans. Inf. Forensic Secur., vol. 15, pp. 2440–2452, Jan. 2020.

[4] Y. Chen, X. Ma, C. Tang, and M. H. Au, “PGC: Decentralized confidential

payment system with auditability,” in Computer Security, Eds. Cham, Switzerland:

Springer, 2020, pp. 591–610.

[5] W. Li, Y. Wang, L. Chen, X. Lai, X. Zhang, and J. Xin, “Fully auditable privacy-

preserving cryptocurrency against malicious auditors,” Cryptology ePrint Archive,

2019. [online]. Available: https://eprint.iacr.org/2019/925.pdf.

[6] E. Androulaki, J. Camenisch, A. D. Caro, M. Dubovitskaya, K. Elkhiyaoui, and B.

Tackmann, “Privacy-preserving auditable token payments in a permissioned

blockchain system,” in Proc. 2nd ACM Conf. Adv. Financial Technol., New York,

NY, USA: Association for Computing Machinery, 2020, pp. 255–267.

[7] Z. Fu, P. Dong, S. Li, and Y. Ju, “An intelligent cross-border transaction system

based on consortium blockchain: A case study in Shenzhen, China,” PLoS ONE

vol. 16, no. 6, p. e0252489, June 2021.

[8] P. Chatzigiannis and F. Baldimtsi, “MiniLedger: Compact-sized anonymous and

auditable distributed payments,” in Computer Security (LNCS), vol. 12972,

Springer, Cham, Sep. 2021, pp. 407– 429.